

REMARKS

1. Present Status Of The Claims

Claims 1-27 are pending in the Application. The Applicant presents these remarks that he believes overcome the rejection of claims.

2. Summary Of The Rejections

The Office Action maintains its rejections claims 1-7 and 9-27 under 35 U.S.C. § 103(a) as being unpatentable over Microsoft FrontPage, Screen Shots, 12/31/99, pp. 1-20 in further view of U.S. Patent No. 6,538,673 (Maslov). Claim 8 remains rejected over the same references, in further view U.S. Patent No. 5,574,898, in further view of Leblang.

2. Request For Interview

Applicant formally requests a telephone interview to discuss the case in accord the remarks provided below.

3. Response To Rejections Of Claims 1-27 Under 35 U.S.C. § 103(a).

The Office Action maintains the rejections of claims 1-7 and 9-27 under 35 U.S.C. § 103(a) as being unpatentable over Microsoft FrontPage, Screen Shots, 12/31/99, pp. 1-20 in further view of U.S. Patent No. 6,538,673 (Maslov). Claim 8 remains rejected over the same references, in further view U.S. Patent No. 5,574,898 (Leblang). For brevity, the Applicant uses independent claim 1 as illustrative of the response for all of currently pending claims 1-27. Furthermore, the traversal is made with the understanding that claims 2-27 are also patentably distinct over the prior art and may include additional features that, beyond those recited in claim 1, provide further, separate, and independent bases for patentability.

As stated in Applicant's previous Responses, claim 1 requires that the names of each of the folders and subfolders depend from the tag names in the markup language file. The Office Action maintains its rejection of claim 1 by combining Maslov with

Microsoft. The Office Action alleges that this feature is somehow described in column 8 of Maslov. However, this feature is not described or suggested anywhere in Maslov. To illustrate this point, Applicant has reproduced EVERY line of column 8 of Maslov below, wherein there is no mention or suggestion of naming folders and subfolders the same as the tag names in the markupfile:

...that correspond to these target windows are hidden on
instruction from user.

Source Document Tree and DOM

We use tree representation of the source online document in creating the transformation script according to the present invention. In the document tree each logical unit of the document such as paragraph, table, heading, emphasis is represented by a node. Node A is a child on node B if and only if the document fragment represented by node A is directly embedded into document fragment represented by node B.

The most popular implementation of the online document tree model for HTML and XML online documents is Document Object Model (DOM) (see the World Wide Web Consortium (W3C) website for more details). Document Object Model is implemented in modern browsers such as Microsoft Internet Explorer 5.0, Netscape Navigator 5.0, and the like. The preferred embodiment of this invention uses DOM as a source document tree model. Other embodiments of this invention can use different tree models for representing the source document.

FIG. 4 shows partial document tree for the source document 10 from FIG. 2 (complete tree is too big to show it on one page). The root of the tree contains BODY element 10 that represents body of the document. The B (for bold) node 20 represents HTML element B that contains the user-selected document fragment 30 on FIG. 2. The path consisting from tree nodes 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, and 41 leads from the root of the tree to the tree node 20.

Creating the Script

A script that performs online document transformation according to this invention (also called WebTransformer Script, or WTS) is created in the following manner.

A source document is displayed in the first window 10 of FIG. 1. The first window 10 is herein referred to as a source window 10. Transformed (target) document is displayed in a second window 30. The second window 30 is herein referred to as a target window 30. Note that target window may be kept invisible until the script is created.

A user can select a source document fragment by clicking the desired fragment using computer pointing device such as a mouse. Selected source document fragment is highlighted. Then, using keys of a computer keyboard, user can expand or contract the selected fragment. In FIG. 1, a fragment 15 is shown as being selected.

Once the fragment 15 is selected, the user can copy the fragment 15 to the target window 30 by selecting "Copy" user interface command from the graphical menu of commands, and a copied fragment then appears in the target window 30 as a target fragment 31. The user can then proceed, for example, to another online document 20, select a fragment 25 therein and copy it to another target location 32 in the target window 30.

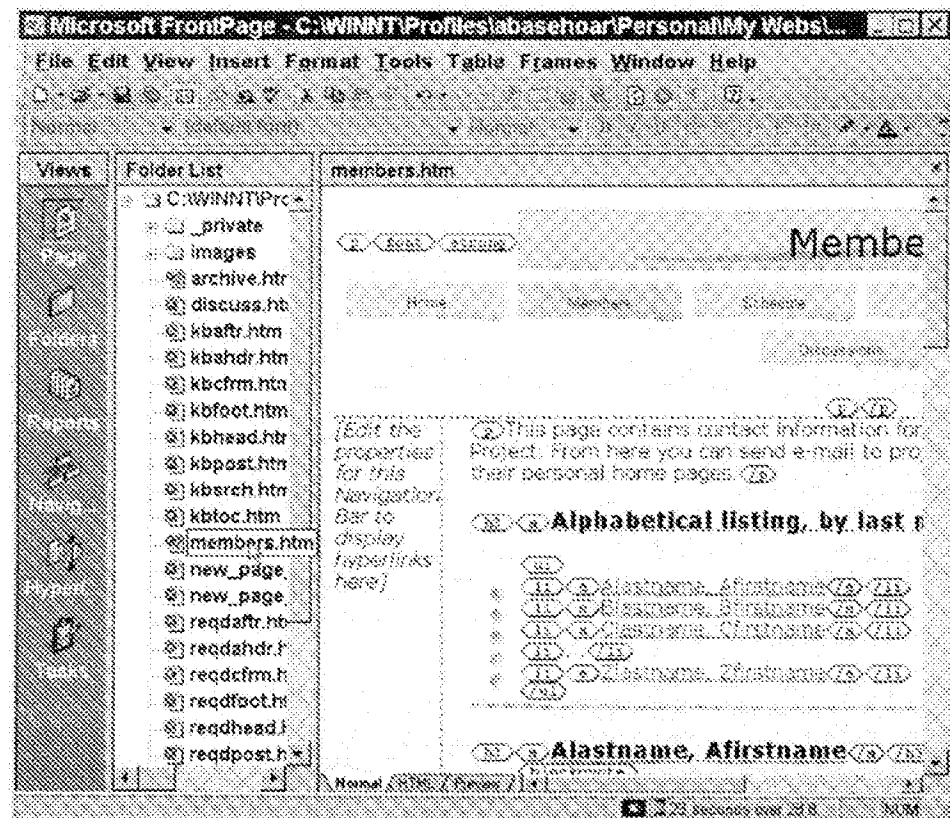
The script that downloads the source document and transforms its fragment into the fragment in the target document is created according to the following rules:

1. Add to the script the "Go To URL" command that causes the browser in the source window to navigate to the source document. The location of the source document includes URL address. The location information can also include additional data that needs to be passed to the

web server to cause displaying of the page selected by user, such as post data and headers.

The command 10 from the sample WebTransformer script shown at FIG. 5 causes the browser to navigate to the corresponding address on the Quicken.RTM. website. This sample script transforms the source document 10 at FIG. 2 to the target document 40.

Maslov describes nothing more than Microsoft in terms of naming the folders, which simply organizes the downloaded html file according to types of content, and not tag names. This is illustrated in the following image from the Microsoft reference:



The Examiner maintains his position that structural elements of a markup language file (i.e., body, table, heading, etc.) somehow equates to random tag names assigned by a markup language file creator who can use ANY tag name. In this regard, a markup language file creator is not limited to structural elements such as “body,” “table,” and “heading.” For example, a tag could be used for the Examiner’s name, “EXAMINER BASEHOAR.” When the system of the present invention operates, a HTML file, for example, would be parsed, and folder names would be created for each of the tag names, including BODY, TABLE, HEADING, TAG1, TAG2EXAMINER BASEHOAR ...TAGN, and the contents of each of those tags in the HTML file would be transferred and loaded into the corresponding subfolders created. This is in sharp contrast to the teachings of the cited art, which does not teach use of tag names in such a random fashion.

Thus, nowhere in Maslov, Microsoft, or any other cited reference, is it disclosed or suggested that tag names are used to name folders. Thus, respectfully, the claims are allowable as they are limited by this feature.

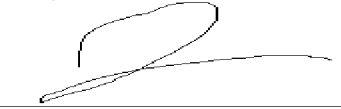
The Applicant respectfully requests an interview with the Examiner to clarify this point.

CONCLUSION

The Applicant has made an earnest and bona fide effort to clarify the issues before the Examiner and to place this case in condition for allowance. In view of the foregoing discussions, it is clear that the cited art, individually or in combination, does not teach all of the elements of any claim of the present invention. Thus, the claimed invention is patentably distinct over the prior art. Therefore, reconsideration and allowance of all of claims 1-27 is believed to be in order, and an early Notice of Allowance to this effect is respectfully requested.

If the Examiner should have any questions concerning the foregoing, the Examiner is invited to telephone the undersigned attorney at (213) 225-2608. The undersigned attorney can normally be reached Monday through Friday from about 9:30 AM to 5:30 PM Pacific Time.

Respectfully submitted,



Dated: March 20, 2007

IVAN POSEY
Reg. No. 43,865
The Chan Law Group, LLP
1055 West 7th Street
Suite 1880
Los Angeles, California 90017
(213) 225-2608

IMP: